

The opinion in support of the decision being entered today  
is *not* binding precedent of the Board.

UNITED STATES PATENT AND TRADEMARK OFFICE

---

BEFORE THE BOARD OF PATENT APPEALS  
AND INTERFERENCES

---

*Ex parte* LUDO GYS

---

Appeal 2006-2723  
Application 09/891,264  
Technology Center 2100

---

Decided: September 6, 2007

---

Before LANCE LEONARD BARRY, MAHSHID D. SAADAT, and  
ALLEN R. MACDONALD, *Administrative Patent Judges*.

SAADAT, *Administrative Patent Judge*.

**DECISION ON REQUEST FOR REHEARING**

Appellant requests reconsideration of our Decision of November 16, 2006, wherein we sustained the Examiner's rejections of the appealed claims under 35 U.S.C. § 103(a).

We have carefully considered the arguments presented by Appellant in the Request and reviewed our decision. However, those arguments do not persuade us that our decision was in error in any respect or we have overlooked any relevant points in reaching our decision that the Examiner properly rejected claims 1-11 under 35 U.S.C. § 103 in view of the applied prior art.

It is Appellant's belief "that the Board has adopted an interpretation of the word 'contain' which is broader than what is reasonable in light of the specification" (Request 2, second ¶). Appellant argues that while a block of executable code may include a block of executable code, it can "contain" a block of executable code only if the word "contain" is read to encompass the concept of "include" (*id.*). Appellant further asserts that such interpretation, when used in combination with the word "container," as recited in the claims, is unreasonably broad (*id.*).

We disagree. The ordinary meaning of "contain" as defined by Merriam-Webster online dictionary<sup>1</sup> is:

2 a: to have within: hold b: comprise, include <the bill contains several new clauses>

(emphasis omitted)

Similarly "contain" is among the synonyms of the word "include:"

---

<sup>1</sup> Merriam-Webster Online, <http://m-w.com/>, Aug. 16, 2007.

synonyms include, comprehend, embrace, involve mean to contain within as part of the whole; include suggests the containment of something as a constituent, component, or subordinate part of a larger whole <the price of dinner includes dessert>.

(emphasis omitted)

Therefore, taking the word “contain” to encompass the concept of “include” is consistent with the ordinary meaning of the terms and not unreasonable for this panel to assert that a block of executable code or a module can “contain” a service independent building block (SIBB). Therefore, consistent with the ordinary meaning of contain and include, a software module of Yates is a container since it contains a SIBB.

Appellant further argues that Yates does not suggest the concept of encapsulation of software building blocks in a software module (Request 3). We note that Appellant merely contends that the software modules of Yates are not containers, but fails to specify what a container is as defined by the instant specification. Appellant’s Specification describes the words “contain” and “container” as:

In the present case the applications executed by the service computer SSC are service containers CONT1 and CONT2 delivered by a service server SSV via the connection CCONL. Additionally, the service computer SSC contains service computer module SCM, containing in the present embodiment a so-called virtual machine, which is a software "execution engine" that safely and compatibly after stringent security

checks executes the byte codes of the service containers on the control means CPUSC. The service computer module SCM may comprise an interpreter decoding and executing statements of the service container's CONT1 program code. Additionally, the service computer module SCM may dynamically link native methods, manage memory or handle exceptions on behalf of the service container CONT1.

(Specification, ¶ linking pp. 7-8.)

Therefore, the applications executed by the service computer are the service containers and the service computer further contains a service computer module which comprises an interpreter decoding and executing statements of the service container's program code. The relationship between these components is an organizational arrangement which represents a hierarchical organization concept. In that regard, there is no particular component that can be considered a "container" except for an application or software modules that contain or include software building blocks.

Similarly, Yates provides for software modules that comprise executable code or codes to be interpreted, which means that the SIBB-type modules still contain individual codes. Although the word "encapsulate" is not explicitly mentioned by Yates, the description of the software module functionality (Yates, col. 3, ll. 5-20) provides for individual software building blocks forming the software module or being contained therein.

We also observe that Appellant equates the word “contain” with “encapsulate” based on sending the service component to the service computer in the same container or another container (Request 4). The reusable software modules of Yates also allow for reconfigurability of the agent by selecting new combination of modules (Abstract). Therefore, the policies that are loaded at run-time and are external to the modules provide for the service component that may be included or encapsulated in any of the modules.

Accordingly, based on the state of the prior art as a whole, we remain of the opinion that it would have been obvious for one of ordinary skill in the art to combine Yates with Beck to provide for the software modules with individual blocks of code contained or included in the modules.

In conclusion, based on the foregoing, we have granted Appellant’s request to the extent that we have reconsidered our decision, but we deny Appellant’s request to make any change therein.

Appeal 2006-2723  
Application 09/891,264

No time period for taking any subsequent action in connection with this appeal may be extended under 37 C.F.R. § 1.136(a)(1)(iv).

DENIED

tdl/gw

SUGHRUE, MION, ZINN MACPEAK & SEAS, PLLC  
2100 PENNSYLVANIA AVENUE, N.W.  
WASHINGTON, DC 20037-3213